

# Haptic Rendering: Introductory Concepts

Kenneth Salisbury <sup>†</sup>, Federico Barbagli <sup>†\*</sup>, Francois Conti <sup>†</sup>

<sup>†</sup> Stanford Robotics Lab - Stanford University - Stanford, CA, U.S.A.

\* Dipartimento di Ingegneria dell'Informazione - University of Siena - Siena, Italy

## 1. Introduction

In the last decade we have seen an enormous increase in interest in the science of haptics. The quest for better understanding and use of haptic abilities (both human and non-human) has manifested itself in heightened activity in disciplines broadly ranging from robotics and telerobotics, to computational geometry and computer graphics, to psychophysics, cognitive science and the neurosciences.

### 1.1 What Is Haptics?

Our use of the word “haptics” broadly refers to touch interactions (physical contact) that occur for the purpose of perception and/or manipulation of objects. These interactions may be between a human hand and a real object, between a robot end-effector and a real-object, between a human hand and a simulated object (via “haptic interface devices” discussed below) or a variety of combinations of human and machine interactions with real, remote and/or virtual objects.

In the early 20th century the word haptics (from the Greek *haptesthai* meaning to “touch”) was used by psychophysicists to label the sub-field of their studies that addressed human touch-based perception and manipulation. In the '70s and '80s significant research efforts in a completely different field, robotics, also began to focus on manipulation and perception by touch. Initially concerned with building autonomous robots, these researchers soon found that building a so-called “dexterous” robotic hand was a problem of much greater complexity and subtlety than their initial naive hopes had suggested. In time these two communities, one that sought to understand the human hand and one that aspired to create devices with dexterity inspired by human abilities, found fertile mutual interest in topics such as sensory design and processing, grasp control and manipulation, object representation and encoding of haptic information, and grammars for describing physical tasks.

In the early '90s a new usage of the word haptics began to emerge [37]. With the confluence of a number of emerging technologies virtualized haptics, or “computer haptics” [43], became possible. Much like computer graphics, computer haptics enabled the display of simulated objects to humans in an interactive manner - however, in this case using a display technology made objects that could be physically palpated. This new sensory display modality presents information by exerting controlled forces on the human hand through a haptic interface (rather than, as in computer graphics, via light from a visual display device impinging on the user’s eye). These forces depend on the physics of mechanical contact (rather than on how light interacts with surfaces). The characteristics of interest in these forces depend on the response of the sensors in the human hand and other body parts (rather than on the eye’s sensitivity to brightness, color, motion, etc.). However, *unlike* computer graphics, haptic interaction is bi-directional with energetic and information flows to, as well as from, the user.

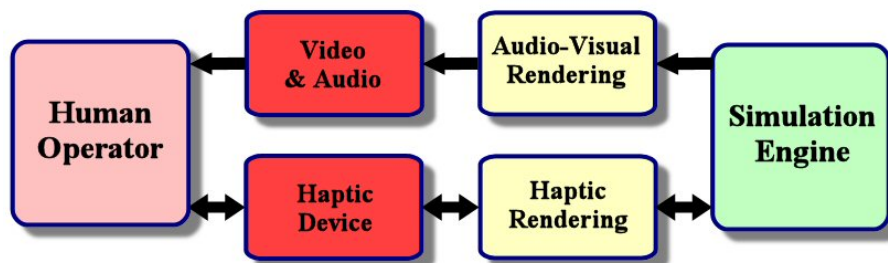
While haptic interaction with simple virtual objects was demonstrated at least as early as the '60s (by Knoll at Bell labs) it was not until recently that sufficient technology was available to make haptic interaction with complex computer simulated objects possible. The combination of high-performance force-controllable haptic interfaces (profiting from lessons in robotics), computational geometric modeling and collision techniques (from computer graphics), cost-effective processing and memory (from consumer PC demand), and an understanding of perceptual needs of the human haptic system (from psychophysics) now enables us to assemble computer haptic systems able to display objects of quite sophisticated complexity and behavior. With the commercial availability of three degree-of-freedom haptic interfaces (they number in the thousands world-wide), software tool kits from several corporate and academic sources, and several commercial haptically enabled applications, the field is experiencing rapid and exciting growth.

## 1.2 Preview of this Article

The focus of this issue of CG&A is on “Haptic Rendering.” By rendering we refer to the process by which desired sensory stimuli are imposed on the user in order to convey information about a virtual haptic object. At the simplest level this information is contained in the representation of the physical attributes of the object (shape, mass, elasticity, texture, temperature, impact, vibration, movement and so on). In the same way that a sphere visually rendered with simple shading techniques will look very different from the same sphere rendered with ray tracing techniques, a sphere haptically rendered with a simple penalty function will feel very different from the same sphere rendered with techniques that also convey mechanical textures and friction present on the surface.

As in the days when people were astonished to see their first wire frame computer generated images, people are astonished today to feel their first virtual object. Yet the rendering techniques we use today will someday seem like yesterday’s wire frame displays - the first steps into a vast field. To understand the issues discussed in subsequent articles, we present here a brief overview of haptic systems, and the techniques needed for rendering the way objects feel. This includes a description of the mechanics of haptic interfaces and a simplified view of the system architecture (including haptic display, computational, and software elements). We then discuss basic haptic rendering algorithms - how we decide what force should be exerted and how we deliver it to the human. We conclude with a few comments on applications, opportunities, and needs in the field.

## 2. Architecture of a Virtual Reality Application with Haptic Feedback



**Figure 1. Basic architecture for a virtual reality application. From left to right: the haptic-visual-audio transducers, haptic-visual-audio rendering algorithms, and simulation engine.**

Virtual Reality (VR) applications strive to simulate real or imaginary scenes with which a user can interact and perceive the effects of his/her actions in real time. Ideally the user interacts with the simulation via all five senses; however, today’s typical VR applications rely on a smaller subset, typically vision, hearing, and more recently, touch.

The structure of a VR application, where visual, auditory and haptic feedback co-exist, is illustrated in Fig. 1. The main elements involved are:

- The simulation engine (green box), responsible for computing how the virtual environment behaves over time.
- Visual, auditory and haptic rendering algorithms (yellow boxes), which compute graphic, sound and force responses of the virtual environment towards the user.
- Transducers (the red boxes), which convert visual, audio and force signals from the computer into a form perceivable by the operator.

The human operator (pink box) typically holds or wears the haptic interface device with his/her hand/body and perceives audio-visual feedback from audio (computer speakers, headphones, etc.) and visual (computer screen, head mounted, etc.) displays.

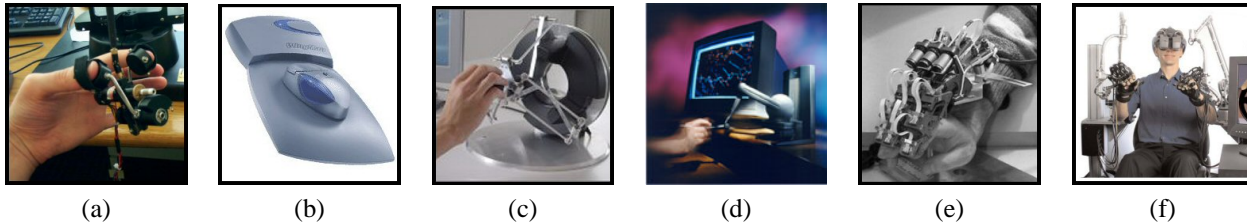
It is important to note that audio and visual channels feature unidirectional information (and energy) flow (from the simulation engine toward the user). This is not the case for the haptic modality, which employs an exchange of information

and energy in two directions, from and toward the user. Bidirectionality, often referred to as the single most important feature of the haptic interaction modality, has some very deep implications that will be described in the rest of the paper.

In the following sections we focus on the modality of haptics in VR based applications, i.e. the lower blocks of the scheme in Fig. 1.

### 3 Examples of Haptic Interface Devices

This section introduces some basic concepts about haptic interface devices that will help the reader throughout the remainder of the text. For a more complete description of the elements that make up such systems the reader is referred to [19].



**Figure 2. A sample of increasingly more complex haptic devices. (a) a force reflecting gripper, (b) the Logitech Wingman force feedback mouse, (c) Forcedimension's OMEGA haptic device, (d) SensAble's Phantom haptic device, (e) the Hand Force Feedback HFF exoskeleton, (f) Immersion's Haptic Workstation**

Haptic interface devices behave like small robots that exchange mechanical energy with a user. We use the term *device-body interface* to highlight the physical connection between operator and device through which energy is exchanged. While such interface can be in contact with any part of the operator's body, hand interfaces have been the most widely used and developed systems to date (see, for instance Fig. 2 (a)-(f)).

One way to distinguish between haptic interface devices is by their grounding locations. For inter-digit tasks, force feedback gloves, such as the HFF [9], read finger-specific contact information and output finger-specific resistive forces, but cannot reproduce object net weight or inertial forces; similar hand held devices are commonly used in the game industry and are built using low-cost vibro-tactile transducers, which produce synthesized vibratory effects. Exoskeleton mechanisms or body-based haptic interfaces present more complex multi-degree-of-freedom motorized devices and are placed on a person's leg or arm. Finally, ground-based devices include force reflecting joysticks and today's desktop haptic interfaces.

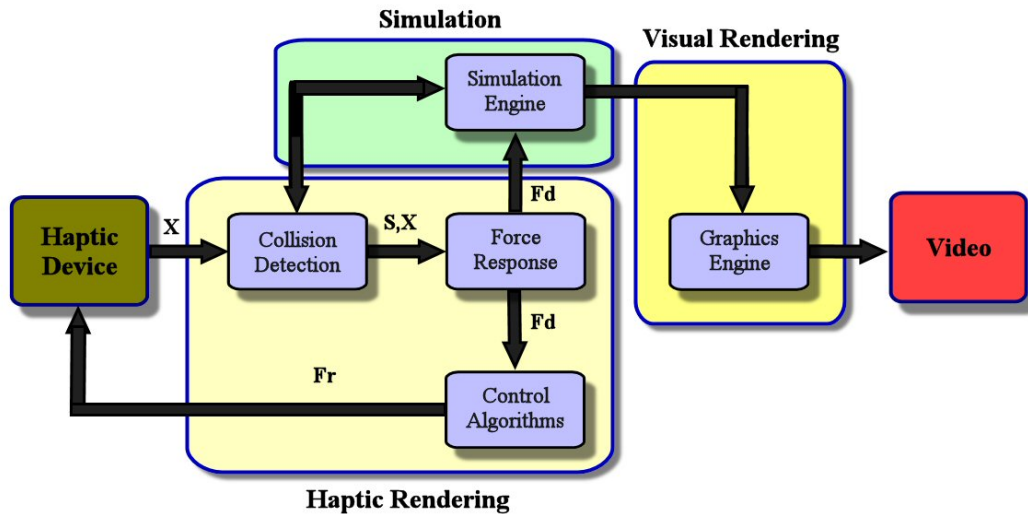
Another distinction between haptic interface devices is their intrinsic mechanical behavior. Impedance haptic devices simulate mechanical impedance, i.e. read position and send force. Admittance haptic devices simulate mechanical admittance, i.e. read force and send position. Simpler to design and much cheaper to produce, impedance type architectures are today the most commonly found. Admittance based devices, i.e. the Haptic Master [46], are generally used for application where high forces in a large workspace are required. In the following we will focus our attention only on impedance haptic devices.

Haptic interface devices are also classified by considering the number of degrees of freedom (DOF) of motion (or force) present at the device-body interface, i.e. the number of dimensions that characterize the possible movements/forces exchanged between device and operator. Each degree of freedom can be passive or actuated, sensed or not sensed.

Some characteristics that are commonly considered to be desirable for haptic interface devices are: low back-drive inertia and friction, and minimal constraints on motion imposed by the device kinematics, so that free motion feels free. Isotropy of useful - that is to say that having inertia, friction, stiffness, and resonate frequency properties balanced in all directions regularizes the device so that users don't have to (unconsciously) compensate for parasitic forces. Similarly the range, resolution, and bandwidth of position sensing and force reflection need to be balanced. Finally, proper ergonomics enable the human operator feel to focus when wearing or manipulating the haptic interface, since pain, or even discomfort, can reduce overall performance due to attention shifts.

In the following we will consider haptic rendering algorithms applicable to single and multiple degree-of-freedom devices. More specifically we will focus on the following classes of impedance-type devices:

- 1 DOF devices such as a haptic knob [41], a haptic scissors [30] or a force reflecting gripper [7] (see Fig. 2 (a)).



**Figure 3. Haptic rendering can be split in three main blocks. Collision detection algorithms provide information about contacts  $S$  occurring between an avatar at position  $X$  and objects in the virtual environment. Force response algorithms return the ideal interaction force  $F_d$  between avatar and virtual objects. Control algorithms return a force  $F_r$  to the user approximating the ideal interaction force to the best of the device's capabilities.**

- 2 DOF devices such as the Pantograph [32] or Logitech's Wingman force feedback mouse (see Fig. 2 (b)).
- 3 DOF devices such as the OMEGA [18] and the PHANTOM [25] haptic devices (see Fig. 2 (c) and (d)).
- 6 DOF devices such as the 6-DOF PHANTOM [11], the 6-DOF DELTA [18], and the Freedom 6 [20] devices.
- > 6 DOF devices such as arm and hand exoskeletons [9] (see Fig. 2 (e) and (f)).

#### 4. System Architecture for Haptic Rendering

Haptic rendering algorithms are responsible for computing the correct interaction forces between the haptic interface representation inside the virtual environment and the virtual objects populating such environment. Moreover, haptic rendering algorithms are in charge of making sure that such forces are correctly rendered on the human operator by the haptic device.

One can define an *avatar* as the virtual representation of the haptic interface, that the user is holding/wearing, through which physical interaction with the virtual environment occurs. Clearly the choice of an avatar is dependent on what is being simulated and on the haptic device's capabilities. The operator controls the avatar's position inside the VE. When contact arises between the interface avatar and the VE, action and reaction forces occur. Such forces are regulated by the type of contact supported by the avatar and by its geometry. Within a given application, as for instance the simulation of a surgical intervention, different avatars can be chosen. The same surgical tool can be treated as a volumetric object exchanging forces and positions with the user in a six-dimensional space, or more simply as a pure point representing the tool's tip, thus exchanging forces and positions in a three-dimensional space.

There are a number of important components that comprise a typical haptic rendering algorithm. Referring to Fig. 3 we identify three main blocks.

Collision detection algorithms detect collisions between the objects and avatars in the virtual environment and yield information about where, when, and ideally to what extent collisions (penetrations, indentations, contact area, etc.) have occurred.

Force response algorithms compute the interaction force between avatars and virtual objects when a collision is detected. This force must approximate as closely as possible the contact forces that would normally arise during contact between real

objects. Force response algorithms typically operate on the avatars' positions, the position of all objects that populate the virtual environment and the state of collision between avatars and virtual objects. Their return values are normally force and torque vectors to be applied at the device-body interface.

Often haptic devices will not be able to apply the exact force computed by the force response algorithms to the user due to hardware limitations. Control algorithms are in charge of commanding the haptic device in such a way that minimizes the error between ideal and applicable force. This can often be difficult due to the discrete-time nature of the haptic rendering algorithms, as will be further explained in Section 6. Control algorithms are fed by desired force and torque vectors to be applied at the device-body interface. Their return value are the actual force and torque vectors that will be commanded to the haptic device.

During a typical haptic loop the following sequence of events occurs:

- The position sensors at the haptic interface device joints are sampled.
- The information collected from each sensor is combined to obtain the position of the device-body interface in cartesian space, i.e. the avatars position inside the virtual environment.
- Position information is used by the collision detection algorithm to find if/where objects/avatars collide and report the degree of penetration or indentation that occurs.
- Interaction forces between avatars and virtual objects with which collision has occurred are computed.
- Interaction forces are passed to the control algorithms which then takes care of applying them on the operator through the haptic device while maintaining a stable overall behavior.

The same interaction forces are then used by the simulation engine to compute their effect on objects in the virtual environment. While there are no firm rules about how frequently these computations need to be repeated, it is common to find that a “servo rate” of 1 KHz is used in today’s applications. This rate seems to be a subjectively acceptable compromise that permits reasonably complex objects to be presented with reasonable stiffness (see section 6 for more details). Higher servo rates can provide crisper contact and texture sensations, but only at the expense of reduced scene complexity (or more expensive computers).

In the following sections we will explain the basic principles of haptic rendering algorithms paying particular attention to force response algorithms. While the ability to detect collisions is an important aspect of computing contact force response, we will not dwell on this topic here given the familiarity of the readership of CG&A with the topic. The geometric problem of efficiently detecting when and where contact and inter-object penetrations occur continues to be an important research topic in haptics as well as related fields. The (faster) real-time demands needs of haptic rendering demand more algorithmic performance. One solution is to simply accept less accuracy and use simpler collision model geometries. Alternately, researchers are now finding ways to adapt graphics rendering hardware to enable fast real-time collision detection among quite complex objects. A useful survey of collision detection algorithms for haptics can be found at [23].

## **5. Computing Forces in Response to Contact**

Humans perceive contact with real objects through sensors (receptors) that are located in their skin, joints, tendons and muscles [42]. A simple distinction between the information that can be acquired using these two types of sensors can be made. Tactile information refers to the information acquired through the sensors in the skin with particular reference to the spatial distribution of pressure, or more generally, tractions, across the contact area. Kinesthetic information refers to the information acquired through the sensors in the joints. Interaction forces are normally perceived through a combination of these two. Tool-based interaction paradigm provides a convenient simplification since only forces resulting from contact between the tool’s avatar and environment objects must be rendered. Thus haptic interfaces frequently utilize a “tool handle” physical interface for the user.

To provide a haptic simulation experience our systems are designed to recreate the contact forces that a user would perceive when touching a real object. The haptic interfaces measure the position of the user, in order to know if/when contacts take place, and to collect information needed to determine the correct interaction force. While determining user motion is easy, determining appropriate display forces is a complex process and a subject of much research. Current haptic technology is good, for simple cases, at simulating interaction forces, but is limited when it comes to tactile feedback. In the following we will focus our attention on force response algorithms for rigid objects. Compliant object response modeling adds an extra

dimension of complexity due to non-negligible deformations, the potential for self collision and the general complexity of modeling potentially large and varying areas of contact.

We will distinguish between two types of forces: those due to objects' geometry and those due to objects' surface properties, such as texture and friction.

## 5.1 Force Rendering Algorithms that Depend on Geometry

These force rendering algorithms aspire to recreate the force interaction that a user would feel when touching a frictionless and textureless object. Such interaction forces depend on the geometry of the object being touched, on its compliance and on the geometry of the avatar that represents the haptic interface inside the VE. Typically the number of degrees of freedom necessary to describe the interaction forces between an avatar and virtual object is chosen in order to match the number of actuated degrees of freedom of the haptic device being used (exceptions exist as explained in [6]). Thus for simpler devices, such as a one DOF force reflecting gripper (see Fig. 2 (a)), the avatar will be made up of a couple of points that can only move and exchange forces along the line connecting them. In this case, the force rendering algorithm will compute a simple one DOF squeeze force between the index finger and the thumb, similar to the force that one would feel when cutting an object with a pair of scissors. When using a six DOF haptic device, such as a six DOF PHANTOM or six DOF DELTA devices, the avatar can be an object of any shape. In this case the force rendering algorithm is in charge of computing all the interaction forces between such object and the VE and then apply the resultant force and torque vectors to the user through the haptic device.

In the following we will present the state of the art for force rendering algorithms. The algorithms will be grouped in sets all featuring an equal number of DOF for interaction force.

### 5.1.1 One DOF Interaction

Single-degree-of-freedom devices are capable of measuring the position from and applying forces to the operator along only one spatial dimension. Examples of one DOF interaction are opening a door using a knob, that is constrained to rotate around one axis, squeezing a pair scissors to cut a piece of paper or pressing on a syringe's piston when injecting a liquid inside a patient. One DOF interaction may seem, at a first glance, limited. However, many interesting effects can be rendered which turn out to be extremely useful.

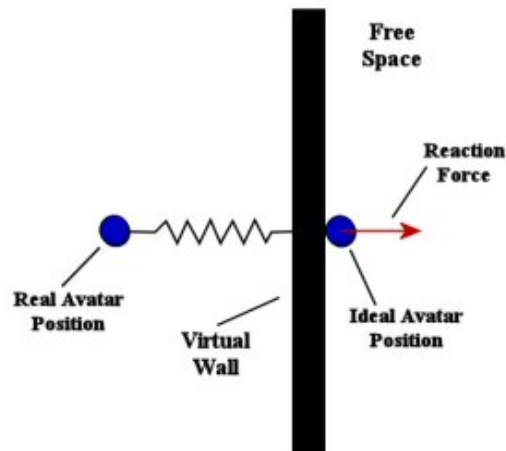


Figure 4. An example of 1 DOF interaction: the virtual wall concept.

Rendering a *virtual wall*, i.e. creating the interaction forces that would arise when contacting an infinitely stiff object, is the prototypical haptic task. As one of the most basic forms of haptic interaction, it is often used as a benchmark in studying haptic stability [12, 17, 2]. Due to the discrete-time nature of haptic interaction, the haptic interface avatar is always going to penetrate inside any virtual object. A positive aspect of this is that the information on how much the avatar has penetrated

inside the object can be used to compute the interaction force. However, some unrealistic effects, such as vibrations in the force values, may arise due to this penetration, as explained in Section 6. Referring to Fig. 4, if we assume that the avatar moves along the  $X$ -axis, and we suppose that the wall is described by  $x < x_W$ , the simplest algorithm to render a virtual wall is given by

$$F = \begin{cases} 0 & x > x_W \\ K(x_W - x) & x \leq x_W \end{cases} \quad (1)$$

where  $K$  represents the wall's stiffness and thus it is ideally very large.

More interesting effects can be accomplished for one DOF interaction. Good examples can be found in [41, 30].

### 5.1.2 Two DOF Interaction

Examples of two DOF interaction exist in everyday life - consider using a mouse to interact with your PC, something you may be doing as you read this paper. It is a bit less intuitive to imagine using two DOF interfaces to interact with three-dimensional objects. This is quite possible and has been shown to be an effective way to interact with simpler three dimensional virtual environments while limiting costs and complexity of haptic devices needed to render the interactions [33]. Two DOF rendering of three dimensional objects, in some cases, can be thought of as pushing a small ball over the surface of a three dimensional object under the influence of gravity [39]. In order to do this, various techniques have been developed to project the ideal three DOF point-contact interaction force on a plane [28, 39], or by evaluating the change of height between two successive contact points on the same surface [21].

### 5.1.3 Three DOF Interaction

Arguably one of the most interesting events that happened in the history of haptics was the recognition, at the beginning of the '90s, that point interaction was an interesting and useful paradigm. This geometric simplification of the general 6DOF problem assumes that we interact with the virtual world with a point probe, and requires that we only compute the three interaction force components at probe's tip. This greatly simplifies the interface device design and makes collision detection and force computation much simpler. Yet, even in this seemingly simple case, we find an incredibly rich array of interaction possibilities and the opportunity to address the fundamental elements of haptics unencumbered by excessive geometric and computational complexity.

In order to compute force interaction with 3D virtual objects, information about how much the probing point, or avatar, has penetrated inside the object is used, as seen in the one DOF case. However, while for one DOF interaction the direction of force is usually trivial, this is not the case for three DOF interaction.

In the case of virtual objects represented using triangular meshes various approaches can be used. *Vector field methods* use a one-to-one mapping between position and force. While these methods can work well in many cases, they have the fundamental limitation of not keeping a history of past avatar positions. This creates problems of determining the direction of the interaction force when dealing with small objects or objects with very complex shapes, as explained in [49, 34]. An interesting example of this is the interaction with an object with very limited width, such as a piece of sheet metal. Non-zero penetration of avatars inside virtual objects can result in their complete crossing through such a thin virtual surface before any force response is computed (e.g. an undetected collision occurs).

In order to address the problems posed by vector field methods Zilles et al. and Ruspini et al. independently introduced the *god-object algorithm* [49] and the *proxy algorithm* [34]. The principle upon which these algorithms are based is the same. While avatars cannot be stopped from penetrating inside virtual objects, additional variables can be used to track a physically realistic contact on the surface of the object. This additional avatar has been called a god-object or proxy. Placing a spring between avatar position and god object/proxy creates a realistic force feedback to the user. In free space, the haptic interface avatar and the god-object/proxy are co-located and thus no force is returned to the user. When colliding with a virtual object the new god-object/proxy position is found using a two-step process. First a set of active constraints is found. Second the new position of the god-object/proxy is found, starting from the old one, as the closest point to the avatar position constrained to be on all its active constraints.

Both of the above algorithms were successively refined by Morgenbesser et al. with the introduction of the concept of *force shading* [29] which can be seen as the haptic equivalent of Phong shading. While the case of graphic rendering interpolated normals serve to obtain more smooth-looking meshes, in the case of haptic rendering they serve to obtain smooth-changing forces throughout an object's surface.

An interesting variation of the god object/proxy algorithms that can be applied to the case of triangular meshes based on very large quantities of polygons was recently proposed by Walker et al. in [47].

In the case of virtual objects based on implicit surfaces with an analytical representation an extension of the god-object algorithm was introduced by Salisbury et al. in [38]. While the basic idea is the same some differences exist: in the case of implicit surfaces, in fact, collision detection is much faster and many of the variables that are necessary for computing the interaction force, such as its direction and intensity, can be computed using closed analytical forms. Other examples of 3DOF interaction include algorithms for interaction with NURBS-based objects [44, 45] and with Voxels-based objects [4].

#### 5.1.4 More than Three DOF Interaction

The point interaction metaphor has proven to be surprisingly convincing and useful. However, it has limitations. Simulating interaction between a tool's tip and a virtual environment means that no torques can be applied through the contact. This can lead to unrealistic scenarios such as a user feeling the shape of a virtual object through the tip of a tool while the rest of the tool lies inside the object.

In order to improve on this situation, avatars have been used that enable exertion of forces and/or torques with more than three degrees of freedom. Borrowing terminology from the robotic-manipulation community, Barbagli et al. [7] developed an algorithm to simulate 4DOF interaction through soft-finger contact, i.e. a point contact with friction that can support moments (up to a torsional friction limit) about the contact normal. This type of avatar becomes particularly handy when using multiple-point interaction in order to grasp and manipulate virtual objects.

Five DOF interaction, such as the one between a line segment and a virtual object, was implemented by Basdogan et al. [8] to approximate contact between long tools and virtual environments. Moreover, this *ray-based* rendering technique can be used to simulate the interaction of tools that can be modeled as a set of connected line segments and a virtual object.

Algorithms providing for full six DOF interaction forces have been demonstrated by a number of researchers. For example, McNeely et al. [26] simulated interaction between modestly complex rigid objects within an arbitrarily complex environment of static rigid objects represented by voxels and Ming et al. [31, 22] simulated contact between very complex polygonal environments and haptic probes.

## 5.2 Force Rendering Algorithms That Depend on Surface Properties

All real surfaces contain very small irregularities or indentations. It is obviously impossible to distinguish each irregularity when sliding a finger over an object. However, the tactile sensors contained in the human skin are capable of feeling their combined effects when rubbed against a real surface. While in this paper we do not focus on tactile displays, we will briefly present the state of the art for algorithms that are capable of rendering haptic textures and friction properties of virtual objects.

Micro-irregularities act as obstructions when two surfaces slide against each other and generate forces tangential to the surface and opposite to motion. Friction, when viewed at the microscopic level, is a very complicated phenomenon. Nevertheless, simple empirical models exist, such as the one originally proposed by Leonardo da Vinci and later developed by Charles Augustin de Coulomb in 1785. Such models were used as a basis for the simpler frictional models in three degrees of freedom [49, 34, 36]. Outside of the haptic community many models to render friction with higher accuracy have been developed in the past, such as the Karnopp model for modeling stick-slip friction, the Bristle model and the Reset Integrator model. Higher accuracy, however, comes at a price, namely speed, which is a critical factor in real-time applications. Any choice of modeling technique must take this trade-off into account. More accurate haptic rendering algorithms for friction have been developed in the past keeping this trade-off in mind (see for instance [16]).

Real surfaces are generally covered by some type of texture or pattern. In order to render the forces that would be generated when touching such textures various techniques have been proposed in the past, many of which find their inspiration in analogous techniques used in modern computer graphics. Thus, while in computer graphics texture mapping can be used for adding realism to computer-generated scenes by projecting a bitmap image onto surfaces being rendered, the same can be done haptically. This was first proposed in a 2D case by Minsky [28] and later extended to a 3D case by Ruspini et al. [34]. Mathematical functions have also been used to create synthetic patterns. Basdogan et al. [8] and Costa et al. [14] investigated the use of fractals to model natural textures while Pai et al. [40] used a stochastic approach.



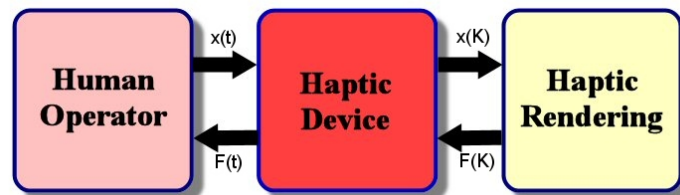
## 6. Controlling Forces Delivered Through Haptic Interfaces

So far we have focused our attention on the algorithms that compute the ideal interaction forces between the haptic interface avatar and the virtual environment. Once such forces have been computed they must be applied to the user. Due to limitations of haptic device technology, however, it may be impossible at times to apply the exact value of the force as computed by force rendering algorithms.

Various issues contribute to limiting the capability of a haptic device to render a desired force or, more often, a desired impedance:

- Haptic interfaces can only exert forces with limited magnitude and not equally well in all directions - thus rendering algorithms must ensure that no output components saturate, as this would lead to erroneous or discontinuous application of forces to the user.
- Haptic devices are not ideal force transducers. An ideal haptic device should be able to render zero impedance when simulating movement in free space, and any finite impedance when simulating contact with an object featuring such impedance characteristics. This is usually not the case due to the friction, inertia and backlash present in most haptic device.
- Haptic rendering algorithms operate in discrete time while users operate in continuous time (see Fig. 5). While moving into and out of a virtual object, the sampled avatar position will always lag behind the actual continuous-time position of the avatar. This implies that as one presses on a virtual object, one needs to perform less work than in reality, while, as one lets go, one has more work returned by the virtual object than would have been returned by its real-world counterpart [17]. In other terms, touching a virtual object is a way to extract energy from it. This extra energy can cause unstable response of haptic devices.
- Haptic device position sensors have finite resolution. As a consequence of this there is always quantization error in determining where and when contact occurs. While this error may not be easily perceived by users it can create stability problems.

All of the above issues can limit the realism of a haptic application and are well known to practitioners in the field. The first two issues usually depend more on the mechanics of the devices used while the latter two depend on the digital nature of VR applications. In the following we will present a brief description of the state of the art of control algorithms that ensure stable haptic interaction in the face of such limitations in virtual environments.



**Figure 5. Haptic devices create a closed loop between user and haptic rendering/simulation algorithms.  $x(t)$  and  $F(t)$  are continuous-time position and force signals exchanged between user and haptic device.  $x(K)$  and  $F(K)$  are discrete-time position and force signals exchanged between haptic device and virtual environment.**

### 6.1 How To Obtain Stable Haptic Interaction

As mentioned in section 2, haptic devices, in contrast to displays that address other human senses, feature a bidirectional flow of energy, to and from the user. This creates a feedback loop featuring user, haptic device and haptic rendering/simulation algorithms as shown in Fig. 5. This in turn can create instability.

The problem of stable haptic interaction has received a lot of attention in the last decade (see for instance [12, 17, 2]). The main problem of studying the stability of the loop in Fig. 5 is the presence of the human operator whose dynamic behavior cannot be generalized with a simple transfer function. In order to create robust algorithms capable of working for any user, passivity theory has been largely employed. Referring to the simple case of a virtual wall, such as the one presented in Fig. 4, Colgate was the first to analytically show that a relation exists between the maximum stiffness that can be rendered by a device, its level of mechanical damping, the level of digital damping commanded to the device and the servo-rate used to control the device. More specifically, in order to have stable interaction, the following relationship should hold

$$b > KT/2 + B$$

i.e. the device damping  $b$  should always be higher than the sum of the level of digital damping that can be controlled to the device  $B$  and the product  $KT/2$  where  $K$  is the stiffness to be rendered by the device and  $T$  is the servo-rate period. Stiffer walls will tend to become unstable for higher servo-rate periods. This will result in high frequency vibrations and possibly in uncontrollably high levels of force. Instability can be limited by increasing the level of mechanical damping featured by the device, even though this limits the device's capabilities of simulating null impedance when simulating free-space movements of the device. Thus high servo-rates (or low servo-rate periods) are a key issue for stable haptic interaction.

Various solutions have been proposed in the past to limit the occurrence of unstable behavior for haptic devices. Two main sets of techniques exist:

- Solutions that use virtual damping in order to limit the flow of energy from virtual environment toward user when this could create unstable behavior [13, 2, 27, 35]. Colgate was the first to introduce the concept of *virtual coupling*, which can be seen as a connection between haptic device and virtual avatar consisting of stiffness and damping which effectively limits the maximum impedance that needs to be exhibited by the haptic display. By using a virtual coupling one can create virtual environments featuring unlimited stiffness levels, since the haptic device will always attempt to render only the maximum level set by the virtual coupling. While this ensures stability it effectively does not solve the issue of making a haptic device stably render higher stiffness levels.
- Solutions that attempt to speed up haptic servo-rates by decoupling force response algorithms from other slower algorithms, such as the ones computing collision detection, visual rendering and virtual environment dynamics. This can be accomplished by running all of these algorithms in different threads featuring different servo rates, and by letting the user interact with a simpler local representation of a virtual object [1, 15] at the highest possible rate that can be accomplished on the system. Typically, four main threads can be considered. The visual rendering loop is typically run at rates of up to 30Hz. The simulation thread is ran as fast as possible congruently with the overall complexity of the scene being simulated. A slower collision detection thread, which computes a local representation of the part of the virtual object that is closest to the user avatar, is ran at slower rates, in order to limit CPU usage. Finally a faster collision detection and force response is ran at high servo-rates. This is made possible by the fact that the local representation is chosen to be extremely simple (typical examples include planes or spheres). Surface discontinuities are normally not perceived, given that the maximum speed of human movements is normally limited and thus the local representation can always "catch up" with the current avatar position. This approach has gained a lot of success in recent years with the advent of surgical simulators employing haptic devices, since algorithms to accurately compute deformable objects dynamics are still fairly slow and not very scalable [10, 3, 5, 24].

## 7. Conclusions

As haptics moves beyond the buzzes and thumps of today's video games we see technology enabling increasingly believable and complex physical interaction with objects. Already designers can sculpt digital clay figures to rapidly produce new product geometry, museum goers can feel previously inaccessible artifacts, and doctors can train for simple procedures without endangering live patients - all with haptically-enabled commercial products.

Past technological advances that permitted recording, encoding, storage, transmission, editing, and ultimately synthesis of images and sound had profound effects on society. A wide range of human activities, including communication, education, art, entertainment, commerce, and science, were forever changed when we became able to capture, manipulate, and create sensory stimuli nearly indistinguishable from reality. It is not unreasonable to expect that future advancements in haptics will have equally deep effects on our lives. Though today the field is still in its infancy, hints of vast, unexplored intellectual and commercial territory add excitement and energy to a growing number of participants in conferences, courses, product releases, and invention efforts.

For the field to move beyond today's state-of-the-art there are a number of commercial and technological barriers that must be surmounted. Device and software tool oriented corporate efforts have provided the tools needed for the field to begin its steps out of the laboratory, yet there is need for new business models. For example, can we create haptic content and authoring tools that will make the technology broadly attractive? Can the interface devices be made practical and inexpensive enough to make them widely accessible?

Once we move beyond single-point force-only interactions with rigid objects there are a number of technical and scientific avenues that should be explored. Multi-point, multi-hand and multi-person interaction scenarios all offer enticing richness in interactivity. Adding sub-modality stimulation such as tactile (pressure distribution) display and vibration could add subtle but important richness to the experience. Modeling compliant objects, such as for surgical simulation and training, presents many challenging problems to enable realistic deformations, arbitrary collisions, and topological changes caused by cutting and joining (e.g. suturing) actions.

Improved accuracy and richness in object modeling and haptic rendering will require advances in our understanding of how to represent and render psychophysically and cognitively germane attributes of objects, as well as algorithms and perhaps specialty hardware (haptic or physics engines?) to carry out real-time computations.

Development of multi-modal workstations that provide haptic, visual, and auditory engagement will offer opportunities for more integrated interactions. The psychophysical and cognitive details needed to enable successful multi-modality interactions are only beginning to be understood. For example how do we encode and render an object so that there is a seamless consistency and congruence across sensory modalities – does it look like it feels? Are the density, compliance, motion and appearance of the object familiar and unconsciously consistent with context? Is there enough predictability in sensory events that we consider objects to be persistent and are we able to make correct inference about properties?

Finally we should not forget that touch and physical interaction are among the fundamental ways in which we come to understand our world and to effect changes in it. This is true on a developmental level as well as on an evolutionary level. Early man's survival demanded the development of the embodied knowledge needed to reason about the world. Indeed, language itself may stem our immersion in the physics of the world. In order for early primates to survive in a physical world, as suggested by Frank Wilson [48], "... a new physics would eventually have to come into this [their] brain, a new way of registering and representing the behavior of objects moving and changing under the control of the hand. It is *precisely* such a representational system – a syntax of cause and effect, of stories and of experiments, each having a beginning, a middle, and an end – that one finds at the deepest levels of the organization of human language." It may well be that our efforts to communicate information by literal rendering of the way objects feel through haptic technology actually reflect a deeper desire to speak with an inner, physically-based language that has yet to be given a true voice.

## References

- [1] Y. Adachi, T. Kumano, and K. Ogino. Intermediate representation for stiff virtual objects. In *IEEE Virtual Reality Annual Intl. Symposium*, pages 203–210, Research Triangle Park, N. Carolina, March 1995.
- [2] R. Adams and B. Hannaford. Stable haptic interaction with virtual environments. *IEEE Transactions on Robotics and Automation*, 15(3):465–474, 1999.
- [3] O. Astley and V. Hayward. Multirate haptic simulation achieved by coupling finite element meshes through norton equivalents. In *Proceedings of IEEE Conference on Robotics and Automation (ICRA98)*, pages 989–994, Leuven, Belgium, 1998.
- [4] R. S. Avila and L. M. Sobierajski. A haptic interaction method for volume visualization. In *IEEE Proceedings of Visualization 96*, pages 197–204, 1996.
- [5] F. Barbagli, D. Prattichizzo, and K. Salisbury. Dynamic local models for stable multi-contact haptic interaction with deformable objects. In *Haptics Symposium 2003*, pages 109–116, Los Angeles, California.
- [6] F. Barbagli and K. Salisbury. The effect of sensor/actuator asymmetries in haptic interfaces. In *Haptics Symposium 2003*, pages 140–147, Los Angeles, California.
- [7] F. Barbagli, K. Salisbury, and R. Devengenzo. Enabling multi-finger, multi-hand virtualized grasping. In *Proceedings of the IEEE International Conference on Robotics and Automation ICRA 2003*, volume 1, pages 806–815, Taipei, Taiwan, September 2003.
- [8] C. Basdogan, C.-H. Ho, and M. A. Srinivasan. A ray-based haptic rendering technique for displaying shape and texture of 3d objects in virtual environments. In *Proceedings of ASME Dynamic Systems and Control Division*, volume DSC-Vol.61, pages 77–84, 1997. [http://touchlab.mit.edu/publications/1997\\_001.pdf](http://touchlab.mit.edu/publications/1997_001.pdf).
- [9] M. Bergamasco. *Artificial Life and Virtual Reality*, N. Magnenat Thalmann and D. Thalmann editors, JohnWiley, chapter Manipulation and Exploration of Virtual Objects, pages 149–160. 1994.
- [10] M. C. Cavusoglu and F. Tendick. Multirate simulation for high fidelity haptic interaction with deformable objects in virtual environments. In *Proceedings of the IEEE International Conference on Robotics and Automation*, pages 2458–2465, San Francisco, CA, May 2000.

- [11] A. Cohen and E. Chen. Six degree-of-freedom haptic system for desktop virtual prototyping applications. In *Proceedings of the ASME Winter Annual Meeting, Dynamics Systems and Control, DSC-Vol67*, pages 401–402, Nashville, Tennessee, November 1999.
- [12] J. Colgate and J. Brown. Factors affecting the z-width of a haptic display. In *Proceedings IEEE Int. Conf. Robotics and Automation*, pages 3205–3210., Los Alamitos, CA, 1994.
- [13] J. Colgate, M. Stanley, and J. Brown. Issues in the haptic display of tool use. In *International Conference on Intelligent Robots and Systems*, pages 140–145, Pittsburgh, PA, August 1995.
- [14] M. Costa and M. Cutkosky. Roughness perception of haptically displayed fractal surfaces. In *Proc. of the ASME Dynamic Systems and Control Division*, volume 69(2), pages 1073–1079, Orlando, FL, November 2000.
- [15] D. d’Aulignac, R. Balaniuk, and C. Laugier. A haptic interface for a virtual exam of a human thigh. In *Proc. IEEE Int. Conf. Robotics and Automation*, volume 3, pages 2452–2457, 2000.
- [16] P. Dupont, V. Hayward, B. Armstrong, and F. Altpeter. Single state elasto-plastic friction models. *IEEE Transactions on Automatic Control*, 47(5):787–792, 2002.
- [17] B. Gillespie and M. Cutkosky. Stable user-specific haptic rendering of the virtual wall. In *Proceedings of the ASME International Mechanical Engineering Conference and Exposition*, volume 58, pages 397–406, Atlanta, GA, November 1996. ASME.
- [18] S. Grange, F. Conti, P. Helmer, P. Rouiller, and C. Baur. Overview of the delta haptic device. In *Proceedings of Eurohaptics 2001*, Birmingham, England, July 2001.
- [19] V. Hayward and O. Astley. Performance measures for haptic interfaces. In E. Giralt G., Hirzinger G., editor, *1996 Robotics Research: The 7th International Symposium*, pages 195–207. Springer Verlag, 1996.
- [20] V. Hayward, V. Gregorio, P. Astley, O. Greenish, S. Doyon, M. Lessard, L. McDougall, J. Sinclair, I. Boelen, S. Chen, X. Demers, J.-P. Poulin, J. Benguigui, I. Almey, N. Makuc, and B. Zhang. Freedom-7: A high fidelity seven axis haptic device with application to surgical training. In *Experimental Robotics V, Lecture Notes in Control and Information Science 232*, V. Casals and A. T. de Almeida editors.
- [21] V. Hayward and D. Yi. *Change of Height: An Approach to the Haptic Display of Shape and Texture Without Surface Normal*, pages 570–579. Springer Verlag, 2003.
- [22] Y. J. Kim, M. A. Otaduy, M. C. Lin, and D. Manocha. Six-degree-of freedom haptic display using localized contact computations. In *Proceedings of the Tenth Symposium on Haptic Interfaces For Virtual Environment and Teleoperator Systems*.
- [23] M. C. Lin and D. Manocha. *Handbook of Discrete and Computational Geometry, Second Edition*, chapter Collision and Proximity Queries. CRC Press, J. O’Rourke and E. Goodman editors, 2004.
- [24] M. Mahvash and V. Hayward. Passivity-based high-fidelity haptic rendering of contact. In *2003 IEEE International Conference on Robotics and Automation, ICRA 2003*, pages 3722–3728, Taipei, Taiwan, 2003.
- [25] T. Massie and J. K. Salisbury. The phantom haptic interface: A device for probing virtual objects. In *Proceedings of ASME Haptic Interfaces for Virtual Environment and Teleoperator Systems In Dynamic Systems and Control 1994*, volume 1, pages 295–301, Chicago, IL, November 1994.
- [26] W. McNeely, K. Puterbaugh, and J. Troy. Six degree-of-freedom haptic rendering using voxel sampling. In *Proceedings of ACM-SIGGRAPH*, pages 401–408, 1999.
- [27] B. E. Miller, J. Colgate, and R. A. Freeman. Guaranteed stability of haptic systems with nonlinear virtual environments. *IEEE Transactions on Robotics and Automation*, 16:712–719, 2000.
- [28] M. Minsky. *Computational Haptics: The Sandpaper System for Synthesizing Texture for a Force Feedback Display*. M.S. Thesis, Massachusetts Institute of Technology, Cambridge, MA, 1995.
- [29] H. B. Morgenbesser and M. A. Srinivasan. Force shading for haptic perception. In *Proceedings of the ASME Dynamic Systems and Control Division*, volume DSC-Vol.58, pages 407–412, 1996. [http://touchlab.mit.edu/publications/1996\\_003.pdf](http://touchlab.mit.edu/publications/1996_003.pdf).
- [30] A. M. Okamura, R. J. Webster, J. T. Nolin, K. W. Johnson, and H. Jafry. The haptic scissors: Cutting in virtual environments. In *2003 IEEE International Conference on Robotics and Automation*, volume 1, pages 828–833, 2003.
- [31] M. A. Otaduy and M. Lin. Sensation preserving simplification for haptic rendering. In *Proceedings of ACM SIGGRAPH 2003 / ACM Transactions on Graphics*, 2003.
- [32] C. Ramstein and V. Hayward. The pantograph: A large workspace haptic device for a multi-modal human-computer interaction. In *Proceedings of CHI’94, Conference on Human Factors in Computing Systems ACM/SIGCHI Companion*, pages 57–58, April 1994.
- [33] G. Robles-De-La-Torre and V. Hayward. Force can overcome object geometry in the perception of shape through active touch. *Nature*, 412:445–448, 2001.
- [34] D. C. Ruspini, K. Kolarov, and O. Khatib. The haptic display of complex graphical environments. In *Computer Graphics (SIGGRAPH 97 Conference Proceedings)*, pages 345–352. ACM SIGGRAPH, 1997.
- [35] J. H. Ryu and B. Hannaford. Time domain passivity control of haptic interfaces. *IEEE Transactions on Robotics and Automation*, 18:1–10, February 2002.
- [36] S. E. Salcudean and T. D. Vlaar. On the emulation of stiff walls and static friction with a magnetically levitated input/output device. In *Proceedings of the ASME DSC*, volume 55(1), pages 303–309, 1994.
- [37] J. K. Salisbury and M. Srinivasan. Section on haptics. virtual environment technology for training. Technical Report BBN Report No. 7661, The Virtual Environment and Teleoperator Research Consortium (VETREC) affiliated with MIT, 1992. <http://touchlab.mit.edu/publications/1992.001.pdf>.
- [38] K. Salisbury and C. Tar. Haptic rendering of surfaces defined by implicit functions. In *ASME Dyn. Sys. and Control Div.*, volume 61, pages 61–67, 1997.

- [39] Y. Shi and D. K. Pai. Haptic display of visual images. In *Proceedings of IEEE Virtual Reality Annual International Symposium (VRAIS '97)*, pages 188–191, Albuquerque, NM, March 1997.
- [40] J. Siira and D. Pai. Haptic textures – a stochastic approach. In *IEEE International Conference on Robotics and Automation*, pages 557–562, Minneapolis, April 1996.
- [41] S. Snibbe, K. E. MacLean, R. Shaw, J. Roderick, W. L. Verplank, and M. Scheef. Haptic techniques for media control. In *Proceedings of the 14th Annual ACM Symposium on User Interface Software and Technology (UIST 2001)*, volume 3(2), pages 199–208, Orlando, FL, November 2001.
- [42] M. A. Srinivasan. *Virtual Reality: Scientific and Technical Challenges. Report of the Committee on Virtual Reality Research and Development*, chapter Haptic Interfaces. National Academy Press, National Research Council, 1995. [http://touchlab.mit.edu/publications/1995\\_006.pdf](http://touchlab.mit.edu/publications/1995_006.pdf).
- [43] M. A. Srinivasan and C. Basdogan. Haptics in virtual environments: Taxonomy, research status, and challenges. *Computers and Graphics*, 21(4):393–404, 1997.
- [44] P. Stewart, Y. Chen, and P. Buttolo. Direct integration of haptic user interface in cad systems. In *Proceedings of ASME*, volume DSC-Vol. 61, pages 93–99.
- [45] T. V. Thompson, D. D. Nelson, E. C. Cohen, and J. M. Hollerbach. Maneuverable nurbs models within a haptic virtual environment. In *Proceedings of the 6th Annual Symposium on Haptic Interfaces for Virtual Environment and Teleoperator Systems*, volume DSC-Vol. 61, pages 37–44, Dallas, TX, November 1997.
- [46] R. Q. VanderLinde, P. Lammertse, E. Frederiksen, and B. Ruitter. The hapticmaster, a new high-performance haptic interface. In *Proceedings of Eurohaptics 2002*, pages 1–5, Edinburgh, UK, July.
- [47] S. Walker and J. K. Salisbury. Large haptic topographic maps: marsview and the proxy graph algorithm. In *Proceedings of the 2003 ACM symposium on Interactive 3D*, pages 83 – 92, Monterey, California, 2003.
- [48] F. Wilson. *The Hand: How its Use Shapes the Brain Language and Human Culture*. Vintage Books, 1999.
- [49] C. Zilles and J. K. Salisbury. A constraint-based god-object method for haptic display. In *Proc. IEE/RSJ International Conference on Intelligent Robots and Systems, Human Robot Interaction, and Cooperative Robots*, volume 3, pages 146–151, 1995.